**Lab Exercise 13- Simple Flow test using PyTest in Metaflow**

**1. Flow Definition**

First, let's ensure that the simple\_flow.py has a proper structure. Here's a modified version to include x for your assertion:

**simple\_flow.py**

from metaflow import FlowSpec, step

class SimpleFlow(FlowSpec):

@step

def start(self):

self.x = 1 # Setting x to 1 for testing

self.next(self.add)

@step

def add(self):

self.y = self.x + 4

self.next(self.end)

@step

def end(self):

print(f"The value of x is {self.x} and y is {self.y}")

if \_\_name\_\_ == '\_\_main\_\_':

SimpleFlow()

**2. Test File with PyTest**

Now, let’s create a test file that uses subprocess to run the flow and checks the value of x.

**test\_simple\_flow.py**

import os

from metaflow import Flow

import subprocess

def test\_flow():

# Command to run the flow

cmd = ['python', 'simple\_flow.py', 'run', '--run-id-file', 'test\_id']

subprocess.check\_call(cmd)

# Read the run ID from the file

with open('test\_id') as f:

run\_id = f.read().strip() # Use .strip() to remove any extra whitespace

# Load the flow run and check the value of x

run = Flow('SimpleFlow')[run\_id]

assert run.data.x == 1 # Check if x is indeed 1

**3. Running the Test**

Make sure you have **Metaflow** installed and that both simple\_flow.py and test\_simple\_flow.py are in the same directory. To run the test, execute:

pytest test\_simple\_flow.py

**Explanation of the Test**

* **Subprocess Call**: The test uses subprocess.check\_call to run the simple\_flow.py script, which executes the Metaflow flow.
* **Reading Run ID**: It reads the run ID from the test\_id file, which is created when the flow is run.
* **Loading the Flow**: Using the Flow class, it loads the completed run by the ID.
* **Assertion**: Finally, it asserts that x is equal to 1, verifying that the flow executed correctly.